Using Structured Events to Predict Stock Price Movement:
An Empirical Investigation

Xiao Ding[†][*]**, Yue Zhang**[‡]**, Ting Liu**[†]**, Junwen Duan**[†]

[†]Research Center for Social Computing and Information Retrieval
Harbin Institute of Technology, China
{xding, tliu, jwduan}@ir.hit.edu.cn
[‡]Singapore University of Technology and Design
yue_zhang@sutd.edu.sg

Abstract

It has been shown that news events influence the trends of stock price movements. However, previous work on news-driven stock market prediction rely on shallow features (such as bags-of-words, named entities and noun phrases), which do not capture structured entity-relation information, and hence cannot represent complete and exact events. Recent advances in Open Information Extraction (Open IE) techniques enable the extraction of structured events from web-scale data. We propose to adapt Open IE technology for event-based stock price movement prediction, extracting structured events from large-scale public news without manual efforts. Both linear and nonlinear models are employed to empirically investigate the hidden and complex relationships between events and the stock market. Large-scale experiments show that the accuracy of S&P 500 index prediction is 60%, and that of individual stock prediction can be over 70%. Our event-based system outperforms bags-of-words-based baselines, and previously reported systems trained on S&P 500 stock historical data.

1 Introduction

Predicting stock price movements is of clear interest to investors, public companies and governments. There has been a debate on whether the market can be predicted. The Random Walk Theory (Malkiel, 1973) hypothesizes that prices are determined randomly and hence it is impossible to outperform the market. However, with advances of AI, it has been shown empirically that stock
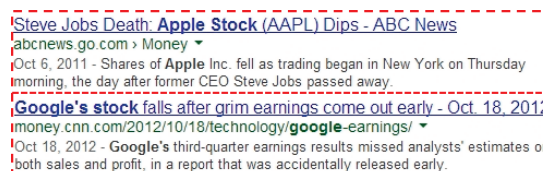

Figure 1: Example news for *Apple Inc.* and *Google Inc.*

price movement is predictable (Bondt and Thaler, 1985; Jegadeesh, 1990; Lo and MacKinlay, 1990; Jegadeesh and Titman, 1993). Recent work (Das and Chen, 2007; Tetlock, 2007; Tetlock et al., 2008; Si et al., 2013; Xie et al., 2013; Wang and Hua, 2014) has applied Natural Language Processing (NLP) techniques to help analyze the effect of web texts on stock market prediction, finding that events reported in financial news are important evidence to stock price movement prediction.

As news events affect human decisions and the volatility of stock prices is influenced by human trading, it is reasonable to say that events can influence the stock market. Figure 1 shows two pieces of financial news about *Apple Inc.* and *Google Inc.*, respectively. Shares of *Apple Inc.* fell as trading began in New York on Thursday morning, the day after its former CEO Steve Jobs passed away. Google's stock fell after grim earnings came out. Accurate extraction of events from financial news may play an important role in stock market prediction. However, previous work represents news documents mainly using simple features, such as bags-of-words, noun phrases, and named entities (Lavrenko et al., 2000; Kogan et al., 2009; Luss and d'Aspremont, 2012; Schumaker and Chen, 2009). With these unstructured features, it is difficult to capture key events embedded in financial news, and even more difficult to model the impact of events on stock market prediction. For example, representing the event "Apple has sued Samsung Electronics for copying 'the look and feel'

---

[*]This work was done while the first author was visiting Singapore University of Technology and Design

of its iPad tablet and iPhone smartphone." using term-level features {"Apple", "sued", "Samsung", "Electronics", "copying", ...} alone, it can be difficult to accurately predict the stock price movements of *Apple Inc.* and *Samsung Inc.*, respectively, as the unstructured terms cannot indicate the actor and object of the event.

In this paper, we propose using structured information to represent events, and develop a prediction model to analyze the relationship between events and the stock market. The problem is important because it provides insights into understanding the underlying mechanisms of the influence of events on the stock market. There are two main challenges to this method. On the one hand, how to obtain structured event information from large-scale news streams is a challenging problem. We propose to apply Open Information Extraction techniques (Open IE; Banko et al. (2007); Etzioni et al. (2011); Fader et al. (2011)), which do not require predefined event types or manually labeled corpora. Subsequently, two ontologies (i.e. VerbNet and WordNet) are used to generalize structured event features in order to reduce their sparseness. On the other hand, the problem of accurately predicting stock price movement using structured events is challenging, since events and the stock market can have complex relations, which can be influenced by hidden factors. In addition to the commonly used linear models, we build a deep neural network model, which takes structured events as input and learn the potential relationships between events and the stock market.

Experiments on large-scale financial news datasets from Reuters[1] (106,521 documents) and Bloomberg[2] (447,145 documents) show that events are better features for stock market prediction than bags-of-words. In addition, deep neural networks achieve better performance than linear models. The accuracy of S&P 500 index prediction by our approach outperforms previous systems, and the accuracy of individual stock prediction can be over 70% on the large-scale data.

Our system can be regarded as one step towards building an expert system that exploits rich knowledge for stock market prediction. Our results are helpful for automatically mining stock price related news events, and for improving the accuracy of algorithm trading systems.

---

[1] http://www.reuters.com/
[2] http://www.bloomberg.com/

2 Method

2.1 Event Representation

We follow the work of Kim (1993) and design a structured representation scheme that allows us to extract events and generalize them. Kim defines an event as a tuple (O_i, P, T), where $O_i \subseteq O$ is a set of *objects*, P is a *relation* over the objects and T is a time interval. We propose a representation that further structures the event to have *roles* in addition to relations. Each event is composed of an **action** P, an **actor** O_1 that conducted the action, and an **object** O_2 on which the action was performed. Formally, an event is represented as $E = (O_1, P, O_2, T)$, where P is the action, O_1 is the actor, O_2 is the object and T is the timestamp (T is mainly used for aligning stock data with news data). For example, the event "Sep 3, 2013 - Microsoft agrees to buy Nokia's mobile phone business for $7.2 billion." is modeled as: (Actor = *Microsoft*, Action = *buy*, Object = *Nokia's mobile phone business*, Time = *Sep 3, 2013*).

Previous work on stock market prediction represents events as a set of individual terms (Fung et al., 2002; Fung et al., 2003; Hayo and Kutan, 2004; Feldman et al., 2011). For example, "Microsoft agrees to buy Nokia's mobile phone business for $7.2 billion." can be represented by {"Microsoft", "agrees", "buy", "Nokia's", "mobile", ...} and "Oracle has filed suit against Google over its ever-more-popular mobile operating system, Android." can be represented by {"Oracle", "has", "filed", "suit", "against", "Google", ...}. However, terms alone might fail to accurately predict the stock price movement of *Microsoft*, *Nokia*, *Oracle* and *Google*, because they cannot indicate the actor and object of the event. To our knowledge, no effort has been reported in the literature to empirically investigate structured event representations for stock market prediction.

2.2 Event Extraction

A main contribution of our work is to extract and use structured events instead of bags-of-words in prediction models. However, structured event extraction can be a costly task, requiring predefined event types and manual event templates (Ji and Grishman, 2008; Li et al., 2013). Partly due to this, the bags-of-words-based document representation has been the mainstream method for a long time. To tackle this issue, we resort to Open IE, extracting event tuples from wide-coverage data with-

out requiring any human input (e.g. templates). Our system is based on the system of Fader et al. (2011) and the work of Ding et al. (2013); it does not require predefined target event types and labeled training examples. Given a natural language sentence obtained from news texts, the following procedure is used to extract structured events:

1. **Event Phrase Extraction.** We extract the predicate verb P of a sentence based on the dependency parser of Zhang and Clark (2011), and then find the longest sequence of words P_v, such that P_v starts at P and satisfies the syntactic and lexical constraints proposed by Fader et al. (2011). The content of these two constraints are as follows:

   - **Syntactic constraint:** every multi-word event phrase must begin with a verb, end with a preposition, and be a contiguous sequence of words in the sentence.
   - **Lexical constraint:** an event phrase should appear with at least a minimal number of distinct argument pairs in a large corpus.

2. **Argument Extraction.** For each event phrase P_v identified in the step above, we find the nearest noun phrase O_1 to the left of P_v in the sentence, and O_1 should contain the subject of the sentence (if it does not contain the subject of P_v, we find the second nearest noun phrase). Analogously, we find the nearest noun phrase O_2 to the right of P_v in the sentence, and O_2 should contain the object of the sentence (if it does not contain the object of P_v, we find the second nearest noun phrase).

An example of the extraction algorithm is as follows. Consider the sentence,

Instant view: Private sector adds 114,000 jobs in July: ADP.

The predicate verb is identified as "adds", and its subject and object "sector" and "jobs", respectively. The structured event is extracted as (*Private sector*, *adds*, *114,000 jobs*).

2.3 Event Generalization

Our goal is to train a model that is able to make predictions based on various expressions of the same event. For example, "Microsoft swallows Nokia's phone business for \$7.2 billion" and "Microsoft purchases Nokia's phone business" report the same event. To improve the accuracy of our prediction model, we should endow the event extraction algorithm with generalization capacity. To this end, we leverage knowledge from two well-known ontologies, WordNet (Miller, 1995) and VerbNet (Kipper et al., 2006). The process of event generalization consists of two steps. First, we construct a morphological analysis tool based on the WordNet stemmer to extract lemma forms of inflected words. For example, in "Instant view: Private sector adds 114,000 jobs in July.", the words "adds" and "jobs" are transformed to "add" and "job", respectively. Second, we generalize each verb to its class name in VerbNet. For example, "add" belongs to the *multiply* class. After generalization, the event (*Private sector*, *adds*, *114,000 jobs*) becomes (*private sector*, *multiply_class*, *114,000 job*). Similar methods on event generalization have been investigated in Open IE based event causal prediction (Radinsky and Horvitz, 2013).

2.4 Prediction Models

1. Linear model. Most previous work uses linear models to predict the stock market (Fung et al., 2002; Luss and d'Aspremont, 2012; Schumaker and Chen, 2009; Kogan et al., 2009; Das and Chen, 2007; Xie et al., 2013). To make direct comparisons, this paper constructs a linear prediction model by using Support Vector Machines (SVMs), a state-of-the-art classification model. Given a training set (d_1, y_1), (d_2, y_2), ..., (d_N, y_N), where $n \in [1, N]$, d_n is a news document and $y_i \in \{+1, -1\}$ is the output class. d_n can be news titles, news contents or both. The output Class +1 represents that the stock price will increase the next day/week/month, and the output Class -1 represents that the stock price will decrease the next day/week/month. The features can be bag-of-words features or structured event features. By SVMs, $y = \arg\max\{Class + 1, Class - 1\}$ is determined by the linear function $\boldsymbol{w} \cdot \boldsymbol{\Phi}(d_n, y_n)$, where $\boldsymbol{w}$ is the feature weight vector, and $\boldsymbol{\Phi}(d_n, y_n)$ is a function that maps d_n into a *M*-dimensional feature space. Feature templates will be discussed in the next subsection.

2. Nonlinear model. Intuitively, the relationship between events and the stock market may be more complex than linear, due to hidden and indirect
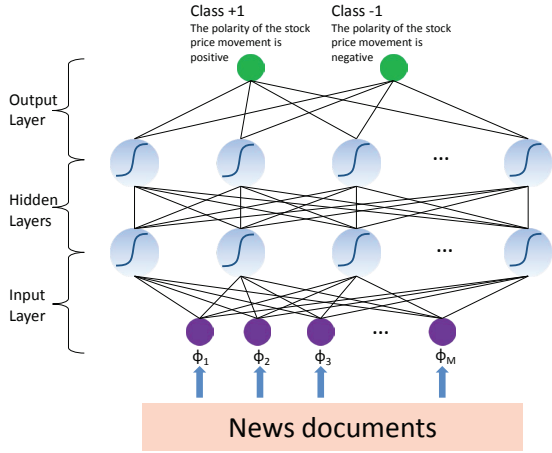
Figure 2: Structure of the deep neural network model

relationships. We exploit a deep neural network model, the hidden layers of which is useful for learning such hidden relationships. The structure of the model with two hidden layers is illustrated in Figure 2. In all layers, the sigmoid activation function σ is used.

Let the values of the neurons of the output layer be y_{cls} ($cls \in \{+1, -1\}$), its input be net_{cls}, and $\mathbf{y_2}$ be the value vector of the neurons of the last hidden layer; then:

$$y_{cls} = f(net_{cls}) = \sigma(\boldsymbol{w_{cls}} \cdot \mathbf{y_2}) \qquad (1)$$

where $\boldsymbol{w_{cls}}$ is the weight vector between the neuron cls of the output layer and the neurons of the last hidden layer. In addition,

$$
\begin{aligned}
y_{2k} &= \sigma(\boldsymbol{w_{2k}} \cdot \mathbf{y_1}) \quad (k \in [1, |y_2|]) \\
y_{1j} &= \sigma(\boldsymbol{w_{1j}} \cdot \boldsymbol{\Phi}(d_n)) \quad (j \in [1, |y_1|])
\end{aligned}
\qquad (2)
$$

Here $\mathbf{y_1}$ is the value vector of the neurons of the first hidden layer, $\boldsymbol{w_{2k}} = (w_{2k1}, w_{2k2}, ..., w_{2k|y_1|})$, $k \in [1, |y_2|]$ and $\boldsymbol{w_{1j}} = (w_{1j1}, w_{1j2}, ..., w_{1jM})$, $j \in [1, |y_1|]$. w_{2kj} is the weight between the kth neuron of the last hidden layer and the jth neuron of the first hidden layer; w_{1jm} is the weight between the jth neuron of the first hidden layer and the mth neuron of the input layer $m \in [1, M]$; d_n is a news document and $\boldsymbol{\Phi}(d_n)$ maps d_n into a M-dimensional features space. News documents and features used in the nonlinear model are the same as those in the linear model, which will be introduced in details in the next subsection. The standard back-propagation algorithm (Rumelhart et al., 1985) is used for supervised training of the neural network.

	train	dev	test
number of instances	1425	178	179
number of events	54776	6457	6593
time interval	02/10/2006 - 18/16/2012	19/06/2012 - 21/02/2013	22/02/2013 - 21/11/2013

Table 1: Dataset splitting

2.5 Feature Representation

In this paper, we use the same features (i.e. document representations) in the linear and nonlinear prediction models, including bags-of-words and structured events.

(1) Bag-of-words features. We use the classic "TFIDF" score for bag-of-words features. Let L be the vocabulary size derived from the training data (introduced in the next section), and $freq(t_l)$ denote the number of occurrences of the lth word in the vocabulary in document d. $\mathbf{TF_l} = \frac{1}{|d|}freq(t_l)$, $\forall l \in [1, L]$, where $|d|$ is the number of words in the document d (stop words are removed). $\mathbf{TFIDF_l} = \frac{1}{|d|}freq(t_l) \times log(\frac{N}{|\{d:freq(t_l) > 0\}|})$, where N is the number of documents in the training set. The feature vector $\boldsymbol{\Phi}$ can be represented as $\boldsymbol{\Phi} = (\phi_1, \phi_2, ..., \phi_M) = (TFIDF_1, TFIDF_2, ..., TFIDF_M)$. The TFIDF feature representation has been used by most previous studies on stock market prediction (Kogan et al., 2009; Luss and d'Aspremont, 2012).

(2) Event features. We represent an event tuple (O_1, P, O_2, T) by the combination of elements (except for T) $(O_1, P, O_2, O_1 + P, P + O_2, O_1 + P + O_2)$. For example, the event tuple (Microsoft, buy, Nokia's mobile phone business) can be represented as (*#arg1*=Microsoft, *#action*=get_class, *#arg2*=Nokia's mobile phone business, *#arg1_action*=Microsoft get_class, *#action_arg2*=get_class Nokia's mobile phone business, *#arg1_action_arg2*=Microsoft get_class Nokia's mobile phone business). Structured events are more sparse than words, and we reduce sparseness by two means. First, verb classes (Section 2.3) are used instead of verbs for *P*. For example, "get_class" is used instead of the verb "buy". Second, we use back-off features, such as $O_1 + P$ ("Microsoft get_class") and $P + O_2$ ("get_class Nokia's mobile phone business"), to address the sparseness of O_1 and O_2. Note that the order of O_1 and O_2 is important for our task since they indicate the actor and object, respectively.
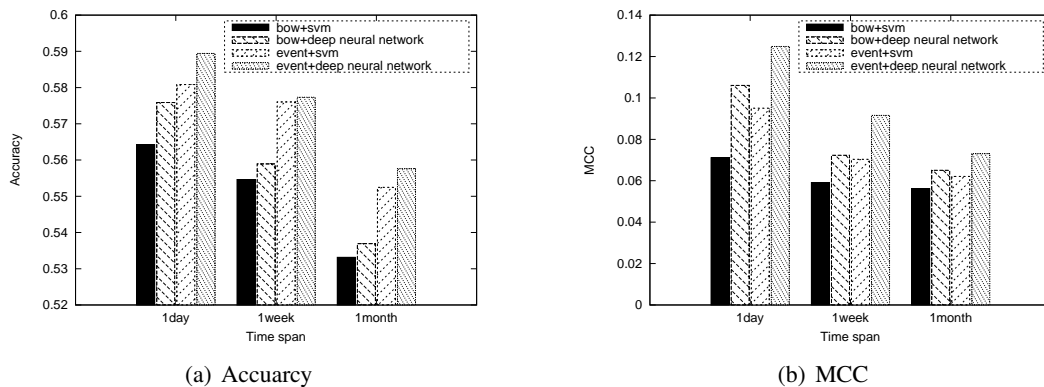
1418

(a) Accuarcy   (b) MCC

Figure 3: Overall development experiment results

3 Experiments

Our experiments are carried out on three different time intervals: short term (1 day), medium term (1 week) and long term (1 month). We test the influence of events on predicting the polarity of stock change for each time interval, comparing the event-based news representation with bag-of-words-based news representations, and the deep neural network model with the SVM model.

3.1 Data Description

We use publicly available financial news from Reuters and Bloomberg over the period from October 2006 to November 2013. This time span witnesses a severe economic downturn in 2007-2010, followed by a modest recovery in 2011-2013. There are 106,521 documents in total from Reuters News and 447,145 from Bloomberg News. News titles and contents are extracted from HTML. The timestamps of the news are also extracted, for alignment with stock price information. The data size is larger than most previous work in the literature.

We mainly focus on predicting the change of the Standard & Poor's 500 stock (S&P 500) index[3], obtaining indices and stock price data from Yahoo Finance. To justify the effectiveness of our prediction model, we also predict price movements of fifteen individual shares from different sectors in S&P 500. We automatically align 1,782 instances of daily trading data with news titles and contents from the previous day/the day a week before the stock price data/the day a month before the stock price data, 4/5 of which are used as the training

---

[3]Standard & Poor's 500 is a stock market index based on the market capitalizations of 500 large companies having common stock listed on the NYSE or NASDAQ.

data, 1/10 for development testing and 1/10 for testing. As shown in Table 1, the training, development and test set are split temporally, with the data from 02/10/2006 to 18/16/2012 for training, the data from 19/06/2012 to 21/02/2013 for development testing, and the data from 22/02/2013 to 21/11/2013 for testing. There are about 54,776 events in the training set, 6,457 events in the development set and 6,593 events in the test set.

3.2 Evaluation Metrics

We use two assessment metrics. First, a standard and intuitive approach to measuring the performance of classifiers is accuracy. However, this measure is very sensitive to data skew: when a class has an overwhelmingly high frequency, the accuracy can be high using a classifier that makes prediction on the majority class. Previous work (Xie et al., 2013) uses an additional evaluation metric, which relies on the Matthews Correlation Cofficient (MCC) to avoid bias due to data skew (our data are rather large and not severely skewed, but we also use MCC for comparison with previous work). MCC is a single summary value that incorporates all 4 cells of a 2*2 confusion matrix (True Positive, False Positive, True Negative and False Negative, respectively). Given TP, TN, FP and FN:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{3}$$

3.3 Overall Development Results

We evaluate our four prediction methods (i.e. SVM with bag-of-word features (bow), deep neural network with bag-of-word features (bow),

1419

		1 day	1 week	1 month
1 layer	Accuracy	58.94%	57.73%	55.76%
	MCC	0.1249	0.0916	0.0731
2 layers	Accuracy	59.60%	57.73%	56.19%
	MCC	0.1683	0.1215	0.0875

Table 2: Different numbers of hidden layers

	title	content	content + title	bloomberg title + title
Acc	59.60%	54.65%	56.83%	59.64%
MCC	0.1683	0.0627	0.0852	0.1758

Table 3: Different amounts of data

Google Inc.					
Company News		Sector News		All News	
Acc	MCC	Acc	MCC	Acc	MCC
67.86%	0.4642	61.17%	0.2301	55.70%	0.1135
Boeing Company					
Company News		Sector News		All News	
Acc	MCC	Acc	MCC	Acc	MCC
68.75%	0.4339	57.14%	0.1585	56.04%	0.1605
Wal-Mart Stores					
Company News		Sector News		All News	
Acc	MCC	Acc	MCC	Acc	MCC
70.45%	0.4679	62.03%	0.2703	56.04%	0.1605

Table 4: Individual stock prediction results

SVM with event features and deep neural network with event features) on three time intervals (i.e. 1 day, 1 week and 1 month, respectively) on the development dataset, and show the results in Figure 3. We find that:

(1) Structured event is a better choice for representing news documents. Given the same prediction model (SVM or deep neural network), the event-based method achieves consistently better performance than the bag-of-words-based method over all three time intervals. This is likely due to the following two reasons. First, being an extraction of predicate-argument structures, events carry the most essential information of the document. In contrast, bag-of-words can contain more irrelevant information. Second, structured events can directly give the actor and object of the action, which is important for predicting stock market.

(2) The deep neural network model achieves better performance than the SVM model, partly by learning hidden relationships between structured events and stock prices. We give analysis to these relationships in the next section.

(3) Event information is a good indicator for short-term volatility of stock prices. As shown in Figure 3, the performance of daily prediction is better than weekly and monthly prediction. Our experimental results confirm the conclusion of Tetlock, Saar-Tsechansky, and Macskassy (2008) that there is a one-day delay between the price response and the information embedded in the news. In addition, we find that some events may cause immediate changes of stock prices. For example, former Microsoft CEO Steve Ballmer announced he would step down within 12 months on 23/08/2013. Within an hour, Microsoft shares jumped as much as 9 percent. This fact indicates that it may be possible to predict stock price movement on a shorter time interval than one day. However, we cannot access fine-grained stock price historical data, and this investigation will be left as future work.

3.4 Experiments with Different Numbers of Hidden Layers of the Deep Neural Network Model

Cybenko (1989) states that when every processing element utilizes the sigmoid activation function, one hidden layer is enough to solve any discriminant classification problem, and two hidden layers are capable to parse arbitrary output functions of input pattern. Here we conduct a development experiment by different number of hidden layers for the deep neural network model. As shown in Table 2, the performance of two hidden layers is better than one hidden layer, which is consistent with the experimental results of Sharda and Delen (2006) on the task of movie box-office prediction. It indicates that more hidden layers can explain more complex relations (Bengio, 2009). Intuitively, three or more hidden layers may achieve better performance. However, three hidden layers mean that we construct a five-layer deep neural network, which is difficult to train (Bengio et al., 1994). We did not obtain improved accuracies using three hidden layers, due to diminishing gradients. A deep investigation of this problem is out of the scope of this paper.

3.5 Experiments with Different Amounts of Data

We conduct a development experiment by extracting news titles and contents from Reuters and Bloomberg, respectively. While titles can give the central information about the news, contents may provide some background knowledge or details. Radinsky et al. (2012) argued that news titles are more helpful for prediction compared to news con-
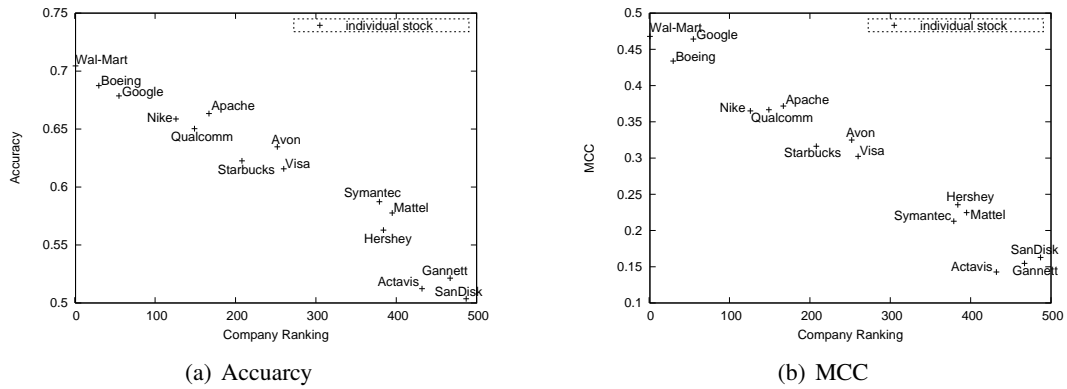
(a) Accuarcy

(b) MCC

Figure 4: Individual stock prediction experiment results

tents, and this paper mainly uses titles. Here we design a comparative experiment to analyze the effectiveness of news titles and contents. First, we use Reuters news to compare the effectiveness of news titles and contents, and then add Bloomberg news titles to investigate whether the amounts of data matters. Table 3 shows that using only news titles achieves the best performance. A likely reason is that we may extract some irrelevant events from news contents.

With the additional Bloomberg data, the results are not dramatically improved. This is intuitively because most events are reported by both Reuters news and Bloomberg news. We randomly select about 9,000 pieces of news documents from Reuters and Bloomberg and check the daily overlap manually, finding that about 60% of the news are reported by both Reuters and Bloomberg. The overlap of important news (news related to S&P 500 companies) is 80% and the overlap of unimportant news is 40%.

3.6 Individual Stock Prediction

In addition to predicting the S&P 500 index, we also investigate the effectiveness of our approach on the problem of individual stock prediction using the development dataset. We select three well-known companies, *Google Inc.*, *Boeing Company* and *Wal-Mart Stores* from three different sectors (i.e. Information Technology, Industrials and Consumer Staples, respectively) classified by the Global Industry Classification Standard (GICS). We use company news, sector news and all news to predict individual stock price movement, respectively. The experimental results are listed in Table 4.

The result of individual stock prediction by us-

ing only company news dramatically outperforms the result of S&P 500 index prediction. The main reason is that company-related events can directly affect the volatility of company shares. There is a strong correlation between company events and company shares. Table 4 also shows that the result of individual stock prediction by using sector news or all news does not achieve a good performance, probably because there are many irrelevant events in all news, which would reduce the performance of our prediction model.

The fact that the accuracy of these well-known stocks are higher than the index may be because there is relatively more news events dedicated to the relevant companies. To gain a better understanding of the behavior of the model on more individual stocks, we randomly select 15 companies (i.e. *Google Inc.*, *Boeing Company*, *Wal-Mart Stores*, *Nike Inc.*, *QUALCOMM Inc.*, *Apache Corporation*, *Starbucks Corp.*, *Avon Products*, *Visa Inc.*, *Symantec Corp.*, *The Hershey Company*, *Mattel Inc.*, *Actavis plc*, *Gannett Co.* and *SanDisk Corporation*) from S&P 500 companies. More specifically, according to the Fortune ranking of S&P 500 companies[4], we divide the ranked list into five parts, and randomly select three companies from each part. The experimental results are shown in Figure 4. We find that:

(1) All 15 individual stocks can be predicted with accuracies above 50%, while 60% of the stocks can be predicted with accuracies above 60%. It shows that the amount of company-related events has strong relationship with the volatility of

---

[4]http://money.cnn.com/magazines/fortune/fortune500/.
The amount of company-related news is correlated to the fortune ranking of companies. However, we find that the trade volume does not have such a correlation with the ranking.

	S&P 500 Index Prediction		Individual Stock Prediction					
			Google Inc.		Boeing Company		Wal-Mart Stores	
	Accuracy	MCC	Accuracy	MCC	Accuracy	MCC	Accuracy	MCC
dev	59.60%	0.1683	67.86%	0.4642	68.75%	0.4339	70.45%	0.4679
test	58.94%	0.1649	66.97%	0.4435	68.03%	0.4018	69.87%	0.4456

Table 5: Final experimental results on the test dataset

company shares.

(2) With decreasing company fortune rankings, the accuracy and MCC decrease. This is mainly because there is not as much daily news about low-ranking companies, and hence one cannot extract enough structured events to predict the volatility of these individual stocks.

3.7 Final Results

The final experimental results on the test dataset are shown in Table 5 (as space is limited, we show the results on the time interval of one day only). The experimental results on the development and test datasets are consistent, which indicate that our approach has good robustness. The following conclusions obtained from development experiments also hold on the test dataset:

(1) Structured events are more useful representations compared to bags-of-words for the task of stock market prediction.

(2) A deep neural network model can be more accurate on predicting the stock market compared to the linear model.

(3) Our approach can achieve stable experiment results on S&P 500 index prediction and individual stock prediction over a large amount of data (eight years of stock prices and more than 550,000 pieces of news).

(4) The quality of information is more important than the quantity of information on the task of stock market prediction. That is to say that the most relevant information (i.e. news title *vs* news content, individual company news *vs* all news) is better than more, but less relevant information.

3.8 Analysis and Discussion

We use Figure 5 to demonstrate our analysis to the development experimental result of *Google Inc.* stock prediction, which directly shows the relationship between structured events and the stock market. The links between each layer show the magnitudes of feature weights in the model learned using the training set.

Three events, (Google, says bought stake in, China's XunLei), (Google, reveals stake in, Chi-

ϕ_1: (Google, says bought stake in, China's XunLei)
ϕ_4: (Google, reveals stake in, Chinese social website)
ϕ_6: (Capgemini, partners, Google apps software)

ϕ_2: (Oracle, sues, Google)
ϕ_5: (Google map, break, privacy law)
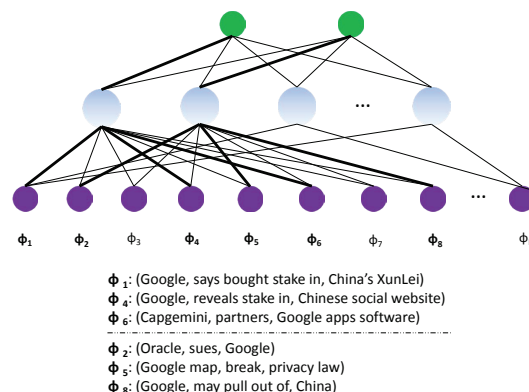ϕ_8: (Google, may pull out of, China)

Figure 5: Prediction of *Google Inc.* (we only show structured event features since backoff features are less informative)

nese social website) and (Capgemini, partners, Google apps software), have the highest link weights to the first hidden node (from the left). These three events indicate that Google constantly makes new partners and expands its business area. The first hidden node has high-weight links to Class +1, showing that Google's positive cooperation can lead to the rise of its stock price.

Three other events, (Oracle, sues, Google), (Google map, break, privacy law) and (Google, may pull out of, China), have high-weight links to the second hidden node. These three events show that Google was suffering questions and challenges, which could affect its reputation and further pull down its earnings. Correspondingly, the second hidden node has high-weight links to Class -1. These suggest that our method can automatically and directly reveal complex relationships between structured events and the stock market, which is very useful for investors, and can facilitate the research of stock market prediction.

Note that the event features used in our prediction model are generalized based on the algorithm introduced in Section 2.5. Therefore, though a specific event in the development test set might have never happened, its generalized form can be found in the training set. For example, "Google acquired social marketing company Wildfire In-

teractive" is not in the training data, but "Google get_class" ("get" is the class name of "acquire" and "buy" in VerbNet) can indeed be found in the training set, such as "Google bought stake in Xun-Lei" on 04/01/2007. Hence although the full specific event feature does not fire, its back-offs fire for a correct prediction. For simplicity of showing the event, we did not include back-off features in Figure 5.

4 Related Work

Stock market prediction has attracted a great deal of attention across the fields of finance, computer science and other research communities in the past. The literature of stock market prediction was initiated by economists (Keynes, 1937). Subsequently, the influential theory of Efficient Market Hypothesis (EMH) (Fama, 1965) was established, which states that the price of a security reflects all of the information available and that everyone has a certain degree of access to the information. EMH had a significant impact on security investment, and can serve as the theoretical basis of event-based stock price movement prediction.

Various studies have found that financial news can dramatically affect the share price of a security (Chan, 2003; Tetlock et al., 2008). Culter et al. (1998) was one of the first to investigate the relationship between news coverage and stock prices, since which empirical text analysis technology has been widely used across numerous disciplines (Lavrenko et al., 2000; Kogan et al., 2009; Luss and d'Aspremont, 2012). These studies primarily use bags-of-words to represent financial news documents. However, as Schumaker and Chen (2009) and Xie et al. (2013) point out, bag-of-words features are not the best choice for predicting stock prices. Schumaker and Chen (2009) extract noun phrases and named entities to augment bags-of-words. Xie et al. (2013) explore a rich feature space that relies on frame semantic parsing. Wang et al. (2014) use the same features as Xie et al. (2013), but they perform non-parametric kernel density estimation to smooth out the distribution of features. These can be regarded as extensions to the bag-of-word method. The drawback of these approaches, as discussed in the introduction, is that they do not directly model events, which have structured information.

There has been efforts to model events more directly (Fung et al., 2002; Hayo and Kutan, 2005;

Feldman et al., 2011). Fung, Yu, and Lam (2002) use a normalized word vector-space to model event. Feldman et al. (2011) extract 9 predefined categories of events based on heuristic rules. There are two main problems with these efforts. First, they cannot extract structured event (e.g. the actor of the event and the object of the event). Second, Feldman et al. (2011) can obtain only limited categories of events, and hence the scalability of their work is not strong. In contrast, we extract structured events by leveraging Open Information Extraction technology (Open IE; Yates et al. (2007); Etzioni et al. (2011); Faber et al. (2011)) without predefined event types, which can effectively solve the two problems above.

Apart from events, sentiment analysis is another perspective to the problem of stock prediction (Das and Chen, 2007; Tetlock, 2007; Tetlock et al., 2008; Bollen et al., 2011; Si et al., 2013). Tetlock (2007) examines how qualitative information (i.e. the fraction of negative words in a particular news column) is incorporated in aggregate market valuations. Tetlock, Saar-Tsechansky, and Macskassy (2008) extend that analysis to address the impact of negative words in all Wall Street Joural (WSJ) and Dow Jones News Services (DJNS) stories about individual S&P500 firms from 1980 to 2004. Bollen and Zeng (2011) study whether the large-scale collective emotion on Twitter is correlated with the volatility of Dow Jones Industrial Average (DJIA). From the experimental results, they find that changes of the public mood match shifts in the DJIA values that occur 3 to 4 days later. Sentiment-analysis-based stock market prediction focuses on investigating the influence of subjective emotion. However, this paper puts emphasis on the relationship between objective events and the stock price movement, and is orthogonal to the study of subjectivity. As a result, our model can be combined with the sentiment-analysis-based method.

5 Conclusion

In this paper, we have presented a framework for event-based stock price movement prediction. We extracted structured events from large-scale news based on Open IE technology and employed both linear and nonlinear models to empirically investigate the complex relationships between events and the stock market. Experimental results showed that events-based document representations are

better than bags-of-words-based methods, and deep neural networks can model the hidden and indirected relationship between events and the stock market. For further comparisons, we freely release our data at http://ir.hit.edu.cn/~xding/data.

Acknowledgments

References

Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction for the web. In *IJCAI*, volume 7, pages 2670–2676.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 5(2):157–166.

Yoshua Bengio. 2009. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127.

Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8.

Werner FM Bondt and Richard Thaler. 1985. Does the stock market overreact? *The Journal of finance*, 40(3):793–805.

Wesley S Chan. 2003. Stock price reaction to news and no-news: Drift and reversal after headlines. *Journal of Financial Economics*, 70(2):223–260.

David M Cutler, James M Poterba, and Lawrence H Summers. 1998. What moves stock prices? *Bernstein, Peter L. and Frank L. Fabozzi*, pages 56–63.

George Cybenko. 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314.

Sanjiv R Das and Mike Y Chen. 2007. Yahoo! for amazon: Sentiment extraction from small talk on the web. *Management Science*, 53(9):1375–1388.

Xiao Ding, Bing Qin, and Ting Liu. 2013. Building chinese event type paradigm based on trigger clustering. In *Proc. of IJCNLP*, pages 311–319, October.

Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam Mausam. 2011. Open information extraction: The second generation. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume One*, pages 3–10. AAAI Press.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics.

Eugene F Fama. 1965. The behavior of stock-market prices. *The journal of Business*, 38(1):34–105.

Ronen Feldman, Benjamin Rosenfeld, Roy Bar-Haim, and Moshe Fresko. 2011. The stock sonarsentiment analysis of stocks based on a hybrid approach. In *Twenty-Third IAAI Conference*.

Gabriel Pui Cheong Fung, Jeffrey Xu Yu, and Wai Lam. 2002. News sensitive stock trend prediction. In *Advances in Knowledge Discovery and Data Mining*, pages 481–493. Springer.

Bernd Hayo and Ali M Kutan. 2005. The impact of news, oil prices, and global market developments on russian financial markets1. *Economics of Transition*, 13(2):373–393.

Narasimhan Jegadeesh and Sheridan Titman. 1993. Returns to buying winners and selling losers: Implications for stock market efficiency. *The Journal of Finance*, 48(1):65–91.

Narasimhan Jegadeesh. 1990. Evidence of predictable behavior of security returns. *The Journal of Finance*, 45(3):881–898.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *ACL*, pages 254–262.

John Maynard Keynes. 1937. The general theory of employment. *The Quarterly Journal of Economics*, 51(2):209–223.

Jaegwon Kim. 1993. *Supervenience and mind: Selected philosophical essays*. Cambridge University Press.

Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. 2006. Extending verbnet with novel verb classes. In *Proceedings of LREC*, volume 2006, page 1.

Shimon Kogan, Dimitry Levin, Bryan R. Routledge, Jacob S. Sagi, and Noah A. Smith. 2009. Predicting risk from financial reports with regression. In *Proc. NAACL*, pages 272–280, Boulder, Colorado, June. Association for Computational Linguistics.

Victor Lavrenko, Matt Schmill, Dawn Lawrie, Paul Ogilvie, David Jensen, and James Allan. 2000. Mining of concurrent text and time series. In *KDD-2000 Workshop on Text Mining*, pages 37–44.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proc. of ACL (Volume 1: Long Papers)*, pages 73–82, August.

Andrew W Lo and Archie Craig MacKinlay. 1990. When are contrarian profits due to stock market overreaction? *Review of Financial studies*, 3(2):175–205.

Ronny Luss and Alexandre d'Aspremont. 2012. Predicting abnormal returns from news using text classification. *Quantitative Finance*, pp.1–14, doi:10.1080/14697688.2012.672762.

Burton G. Malkiel. 1973. *A Random Walk Down Wall Street*. W. W. Norton, New York.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Kira Radinsky and Eric Horvitz. 2013. Mining the web to predict future events. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 255–264. ACM.

Kira Radinsky, Sagie Davidovich, and Shaul Markovitch. 2012. Learning causality for news events prediction. In *Proc. of WWW*, pages 909–918. ACM.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1985. Learning internal representations by error propagation. Technical report, DTIC Document.

Robert P Schumaker and Hsinchun Chen. 2009. Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Transactions on Information Systems (TOIS)*, 27(2):12.

Ramesh Sharda and Dursun Delen. 2006. Predicting box-office success of motion pictures with neural networks. *Expert Systems with Applications*, 30(2):243–254.

Jianfeng Si, Arjun Mukherjee, Bing Liu, Qing Li, Huayi Li, and Xiaotie Deng. 2013. Exploiting topic based twitter sentiment for stock prediction. In *Proc. of ACL (Volume 2: Short Papers)*, pages 24–29, Sofia, Bulgaria, August. Association for Computational Linguistics.

Paul C Tetlock, Maytal Saar-Tsechansky, and Sofus Macskassy. 2008. More than words: Quantifying language to measure firms' fundamentals. *The Journal of Finance*, 63(3):1437–1467.

Paul C Tetlock. 2007. Giving content to investor sentiment: The role of media in the stock market. *The Journal of Finance*, 62(3):1139–1168.

William Yang Wang and Zhenhao Hua. 2014. A semiparametric gaussian copula regression model for predicting financial risks from earnings calls. In *Proc. of ACL*, June.

Boyi Xie, Rebecca J. Passonneau, Leon Wu, and Germán G. Creamer. 2013. Semantic frames to predict stock price movement. In *Proc. of ACL (Volume 1: Long Papers)*, pages 873–883, August.

Alexander Yates, Michael Cafarella, Michele Banko, Oren Etzioni, Matthew Broadhead, and Stephen Soderland. 2007. Textrunner: open information extraction on the web. In *Proc. of NAACL: Demonstrations*, pages 25–26. Association for Computational Linguistics.

Yue Zhang and Stephen Clark. 2011. Syntactic processing using the generalized perceptron and beam search. *Computational Linguistics*, 37(1):105–151.